LUNTRY

Карта безопасности для контейнерных сред

Риски и их митигация для контейнерных приложений и сред на базе оркестратора Kubernetes. Вдохновлено NIST SP 800-190.

версия 1.0

РИСКИ КОНТЕЙНЕРНЫХ СРЕД НА БАЗЕ KUBERNETES

Риски хостовой ОС

- Большая поверхность атаки
  - Использование специализированных ОС для контейнеров
  - Для ОС общего назначения, следование NIST SP 800-123, Guide to General Server Security
- Общее ядро ОС
  - Изолирование приложений разного уровня критичности по разным узлам
  - Исключение запуска не контейнерных приложений на хосте оркестратора
  - AppArmor, SeLinux политики
  - Seccomp профили
  - Альтернативные container runtimes с высокой изоляцией
- Уязвимости компонентов ОС хоста
  - Поддержание системных компонент в актуальном состоянии
  - Управление уязвимостями для системных компонент (kernel, container runtime)
  - Использование хостовой ОС в immutable манере
- Некорректные права доступа пользователей
  - Аудит всех входов в систему
  - Мониторинг аномальных входов
  - Логирование привилегированных действий
  - Обнаружение аномалий
  - Выполнение действий на Node только через специальный operator
  - Доступ на Node через специальный временный привилегированный контейнер
  - Фиксирование всех интерактивных сессий на Node
- Вмешательство в файловую систему ОС хоста
  - Минимизация логики/необходимости пробрасывания файловой системы хоста внутрь контейнера
  - Мониторинг и запрет критичных монтируемых директорий хоста внутрь контейнера

Риски Kubernetes-ресурсов

- Вредоносный YAML
  - Встроенные admission controllers (LimitRange, ResourceQuota и т.д.)
  - Policy Engine (Kyverno, OPA Gatekeeper), Validating Admission Policy
  - Kubernetes Audit Log

Риски контейнеров

- Уязвимости компонентов runtime
  - Управление уязвимостями для компоненты container runtime
- Неограниченный доступ к сети из контейнеров
  - Контроль исходящего трафика контейнеров
  - NetworkPolicy
  - Service Mesh
  - Ingress, Egress gateway
- Небезопасная конфигурация container runtime
  - Автоматизированная проверка конфигураций на соответствие стандартам
  - Использование мандатного контроля доступа (AppArmor, Selinux)
  - Seccomp профили
  - PodSecurityContext, SecurityContext
  - Policy Engine (Kyverno, OPA Gatekeeper), PSA, Validating Admission Policy
- Уязвимости приложений
  - Использование специализированных инструментов для защиты контейнеров
  - Автоматическое профилирование поведения контейнерных приложений
  - Фиксирование процессных, сетевых и файловых аномалий
  - Запуск контейнеров с root filesystems в режиме read-only
- Избыточные привилегии
  - Контроль RBAC
- Отказ в обслуживании
  - Использование ingressClassName
  - Установка запросов и лимитов на ресурсы
- Неавторизованные контейнеры
  - Отдельные окружения для каждой из сред с выделенным контролем
  - Аудит действий с контейнерами
  - Применение инструментов безопасности для контроля соответствия перед использованием
  - Policy Engine (Kyverno, OPA Gatekeeper)
  - Проверка подписи образа
  - Белый список разрешенных registry
  - Единый источник правды (GitOps подход)

Риски интегрированных систем

- Небезопасная конфигурация
  - Аутентификация и авторизация
  - Сетевые политики
  - Настройка в соответствии с лучшими практиками для данного класса решений

Риски образов контейнеров

- Уязвимости образов
  - Использование специализированных инструментов управления уязвимостями
  - Контроль на всем жизненном цикле образа
  - Контроль всех слоев образа
  - Использование quality gates
- Использование недоверенных, неразрешенных, скомпрометированных компонент
  - Компонентный анализ (SBOM)
- Недостатки конфигурации образов
  - Проверка на соответствие лучшим практикам
  - Организация постоянного, централизованного контроля
  - Опциональный запрет на использование образов, не соответствующих требованиям
  - Использование доверенных базовых образов с минимальным наполнением и частым обновлением
  - Исключение наличия инструментов удаленного доступа
- Вредоносный код
  - Проверка образов
- ПО двойного назначения
  - Тонкие образы
- Хранение чувствительной информации
  - Исключение хранения секретов в образах и организация доставки в runtime только по необходимости
  - Использование систем управления секретами
- Небезопасная сборка образов
  - Использование инструментов без доступа к container runtime socket
- Использование недоверенных образов
  - Использование только доверенных источников
  - Проверка цифровой подписи

Риски хранилища образов

- Небезопасное соединение с реестром
  - Взаимодействие только по защищенным каналам
- Устаревшие образы в реестре
  - Очистка хранилища от устаревших, небезопасных образов
  - Использование образов только по неизменяемым именам
- Недостаточные ограничения аутентификации и авторизации
  - Повсеместное использование аутентификации
  - Использование федерации с существующими учётными записями
  - Аудит всех операций записи в хранилище
  - Встраивание контроля и механизмов безопасности в автоматизированный процесс сборки
- Утечка информации
  - Registry Staging
  - Запрет на push из prod кластера

Риски оркестратора

- Неограниченный административный доступ
  - Следование принципу наименьших привилегий
- Неавторизованный доступ
  - Использование MFA для привилегированных пользователей
  - Использование SSO
  - Централизованный аудит доступа PAM
  - Шифрование данных при хранении
- Некорректное разделение межконтейнерного сетевого трафика
  - Сегментация сети и межсервисного взаимодействия
- Смешивание нагрузок разного уровня критичности
  - Изолирование приложений разного уровня критичности по разным узлам
  - Автоматизация шедулинга приложений по выделенным узлам
- Уязвимости системных компонент
  - Обновление системных компонент
  - Использование Policy Engine (Kyverno, OPA Gatekeeper), Validating Admission Policy для митигации
- Небезопасная конфигурация системных компонент
  - Проверка по CIS Kubernetes benchmark
  - Проверка по внутренним требованиям
- Избыточные привилегии пользователей и групп
  - Контроль RBAC
  - Использование Policy Engine (Kyverno, OPA Gatekeeper), Validating Admission Policy
- Доверие оркестратора к Node
  - Возможность аутентификации Node платформой оркестрации
  - Возможность изолирования скомпрометированного Node и удаления из кластера без нарушения или ухудшения общих операций кластера
  - Взаимная аутентификация и сквозное шифрование на уровне сети
  - Следование принципу «secure-by-default»

Luntry – это Комплексная Защита на всем жизненном цикле контейнерных приложений и средств оркестрации на базе Kubernetes

luntry_official · luntrysolution · luntrysolution · info@luntry.ru